

**WHAT IS CLAIMED IS:**

1. An information handling method comprising:  
generating a case set from operational store data including a goal attribute that reflects an outcome to predict and a plurality of predictor attributes aggregated from entity past behavior;  
building a prediction model;  
dynamically caching data during a transaction with an entity;  
computing attribute aggregates in real-time from the dynamically cached data; and  
scoring entities in real-time based on the prediction model, a score being a probability of a modeled outcome and being derived from the same predictor attributes, including the real-time aggregates, which are used in the case set.
2. The method according to Claim 1 further comprising:  
modifying the prediction model in a learning cycle.
3. The method according to Claim 1 further comprising:  
applying business rules in real-time based on model scores.
4. The method according to Claim 1 further comprising:  
tracking transactions with an entity in real-time; and  
generating aggregates real-time based on the tracked entity transactions.
5. A recommender for usage in a data mining system comprising:  
a scoring engine capable of scoring information using at least one data mining model that derives scores from aggregated data;  
an offer manager capable of mapping scores to offers based on entered configuration information; and  
an aggregation engine that computes user-defined aggregates dynamically for real-time scoring by the scoring engine.

6. The recommender according to Claim 5 further comprising:  
an engine that personalizes transactions by applying business rules and prediction models against cached entity data.
7. The recommender according to Claim 5 wherein:  
the aggregation engine computes simple aggregates and compound aggregates..
8. The recommender according to Claim 5 wherein:  
the aggregation engine enables a user to create simple aggregates and compound aggregates by writing Java™ classes.
9. The recommender according to Claim 5 wherein:  
the aggregation engine enables a user to implement complex conditions by writing a Boolean method in Java™.
10. The recommender according to Claim 5 further comprising:  
a recommender server that iteratively tracks correlations discovered in data mining, applies a prediction model to the correlations to determine results, aggregates the results, and data mines the results to identify new aggregates and refine existing aggregates.
11. The recommender according to Claim 5 further comprising:  
a user interface that exposes entity data through a user-written Interface Device Language (IDL), the IDL defining a context object containing the input data including record-sets, singular records, and scalar elements.
12. The recommender according to Claim 11 further comprising:  
a process that reflects on the user interface to access the data elements for aggregation.
13. A recommender comprising:  
an aggregation engine capable of computing aggregates in real-time from in-memory data that is dynamically cached during a session with an entity.

14. The recommender according to Claim 13 wherein:  
the aggregation engine computes a compound aggregate from at least one  
component aggregates.
15. An information handling apparatus comprising:  
a recommender comprising:  
a recommender server capable of reading scoring models, aggregate  
definitions from database tables, and metadata;  
an aggregation engine that accesses information read by the recommender  
server and uses reflection to construct aggregate objects  
incorporating fields, static methods, and aggregate functions so that  
the aggregate objects are ready to compute as requests arrive.
16. The apparatus according to Claim 15 further comprising:  
a scoring engine that computes scores for specific models, makes offers  
contingent on score results, and evokes the aggregation engine to compute  
pertinent aggregates.
17. The apparatus according to Claim 15 further comprising:  
a harness that simulates a recommender environment in a graphical tool used to  
write, compile, and test business rules.
18. The apparatus according to Claim 15 wherein:  
the recommender creates a default business object model the includes facilities of  
the aggregation engine, reducing usage of custom methods in the business  
object model.
19. The apparatus according to Claim 15 further comprising:  
an offer manager that uses criteria entered into a table to select offers based on  
scores from a specified group of offers, the individual offers being  
associated with a selected mining model that predicts likelihood of an  
entity accepting an offer.

20. The apparatus according to Claim 19 wherein:  
attributes of an offer specification used to select the offer include a threshold, a value, and a cost that, in combination, are sufficient to specify selection criteria based on raw score, weighted score, or net profit.
21. The apparatus according to Claim 15 further comprising:  
an interface capable of reloading and recompiling a combination of metadata categories in real-time including aggregate definitions, deployed mining models, business rules, and offer definitions.
22. The apparatus according to Claim 15 further comprising:  
an object reference to the interface, the object reference being a reference to a scaleable server class; and  
an object reference to a management interface, the management object reference being a server instance reference with server instances being invoked individually for management requests.
23. An information handling method comprising:  
reading information including scoring models, aggregate definitions from database tables, and metadata;  
constructing aggregate objects incorporating the information including fields, static methods, and aggregate functions so that the aggregate objects are ready to compute as requests arrive; and  
using aggregate objects to compute aggregates as requests arrive.
24. The method according to Claim 23 further comprising:  
constructing the aggregate objects using reflection.
25. The method according to Claim 23 further comprising:  
computing scores for specific models;  
making offers contingent on score results; and  
evoking the aggregation engine to compute pertinent aggregates.

26. The method according to Claim 23 further comprising:  
simulating a recommender environment in a graphical tool used to write, compile,  
and test business rules.
27. The method according to Claim 23 further comprising:  
using criteria entered into a table to select offers based on scores from a specified  
group of offers, the individual offers being associated with a selected  
mining model that predicts likelihood of an entity accepting an offer.
28. The method according to Claim 27 wherein:  
attributes of an offer specification used to select the offer include a threshold, a  
value, and a cost that, in combination, are sufficient to specify selection  
criteria based on raw score, weighted score, or net profit.
29. The method according to Claim 23 further comprising:  
reloading and recompiling a combination of metadata categories in real-time  
including aggregate definitions, deployed mining models, business rules,  
and offer definitions.
30. An article of manufacture comprising:  
a controller usable medium having a computable readable program code embodied  
therein for data mining, the computable readable program code further  
comprising:  
a code capable of causing the controller to generate a case set from  
operational store data including a goal attribute that reflects an  
outcome to predicts and a plurality of predictor attributes  
aggregated from entity past behavior;  
a code capable of causing the controller to build a prediction model;  
a code capable of causing the controller to compute attribute aggregates in  
real-time; and  
a code capable of causing the controller to score entities in real-time based  
on the prediction model, a score being a probability of a modeled

outcome and being derived from the same predictor attributes,  
including the real-time aggregates, that are used in the case set.